Understanding Adoption and Best Practices of AI Coding Agents on GitHub

Iqra Zafar
Department of Computer Science, University of Engineering and Technology, Taxila
22-cs-75@uettaxila.edu.pk

***Abstract*—AI coding agents are increasingly integrated into software development workflows, yet systematic evidence on who adopts them and which practices yield successful contributions remains limited. This study analyzes 932,791 agent-authored pull requests (Agentic-PRs) from 116,211 GitHub repositories involving 72,189 practitioners. Experienced practitioners (43.3%) are the largest adopters, contradicting assumptions that agents mainly assist novices. High-quality PRs are concise (<150 lines), single-commit, well-tested, and clearly documented. Findings provide empirical guidance for fostering effective human–AI collaboration and improving reproducibility in AI-assisted software engineering.**

I. Introduction

AI coding agents such as Codex, Copilot, and Devin represent a major transformation in how code is produced and reviewed. They autonomously generate and modify source code, contributing to thousands of pull requests (PRs) daily. This paper investigates two core research questions:

- **RQ1a:** Who adopts these agents, and how do adoption patterns vary across repositories and programming ecosystems?
- **RQ1b:** What practices (PR size, task type, commit granularity) correlate with the quality of Agentic-PRs?

II. Related Work

Prior research has examined AI-assisted programming through small-scale user studies and productivity surveys [1], [2]. These studies highlight usability and trust concerns but lack large-scale empirical evidence. Li et al. (2025) introduced the AIDEV dataset to enable systematic investigation of coding agents in real-world repositories. Building on that foundation, our study quantitatively explores adoption and best practices across nearly one million Agentic-PRs, focusing on developer experience, repository maturity, and structural quality factors.

III. Dataset and Methodology

The dataset includes 932,791 PRs from 116,211 repositories by 72,189 practitioners using five agents: OpenAI Codex, Copilot, Cursor, Devin, and Claude Code. Practitioner experience was inferred from GitHub account age; repository maturity from star counts. Quality was measured via a composite score combining merge success (40%), time-to-merge (20%), approval (20%), and change-request rate (20%). Statistical analysis employed Mann-Whitney U tests, Cohen's d, Pearson correlations, and Random-Forest regression for multivariate importance ranking. All scripts, visualizations, and data-processing steps are documented in the accompanying Jupyter notebook (`Q2.ipynb`) for full reproducibility.

TABLE I: Summary of Adoption Patterns (Aggregated from 932k PRs)

Dimension	**Result**	**Interpretation**
Practitioner Experience	43.3% Exp., 21.3% New, 18.2% Jr., 11.7% Interm.	Experienced practitioners dominate; agents act as productivity tools.
Influence (Followers)	79% low (0-10)	Adoption is grassroots rather than influencer-driven.
Language (top 5 of 250)	Python 27.1%, TS 18.5%, JS 15.9%, HTML 8.0%, Go 3.1%	Python steady; Go favored by experienced contributors.
Repository Maturity	96.1% <100 stars (median 0)	Mostly small or personal repositories.
Agent Distribution	Codex 87.3%, Copilot 5.4%, Cursor 3.5%, Devin 3.2%, Claude 0.6%	Codex dominates adoption.
Temporal Trend	Dec 2024–Jul 2025 (peak Jun 2025)	Rapid acceleration in adoption rates.

IV. Results and Discussion

A. *RQ1a: Who Adopts Coding Agents?*

Finding 1: Adoption spans all experience levels but is led by professionals (43.3%), with newcomers representing 21.3%, contradicting the assumption that AI agents primarily serve as learning aids for beginners. Python, TypeScript, and JavaScript ecosystems dominate usage. Adoption patterns vary significantly by repository maturity: 96.1% of Agentic-PRs target repositories with fewer than 100 stars, suggesting agents are primarily used in personal projects and small-scale developments where experimentation and iteration are easier. Experience-language cross-analysis reveals that experienced practitioners favor Go (7.4%), while newcomers prefer HTML (15.9%), indicating ecosystem-specific adoption patterns.

B. *RQ1b: Quality Practices and Correlates*

1) Task Type Correlation with Quality: **Finding 2:** Task type significantly correlates with PR quality. Agents perform

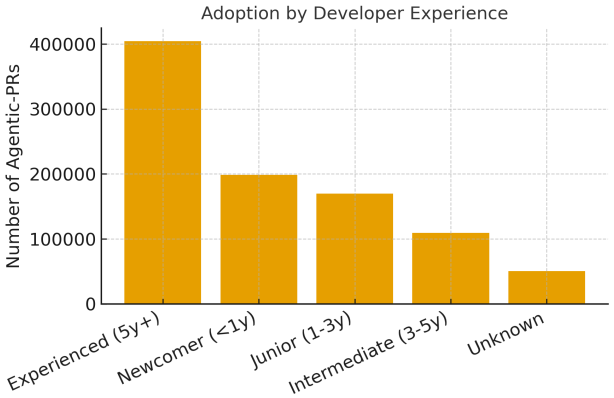

Fig. 1: Adoption by practitioner experience. Experienced (5 y+) practitioners contribute the largest share of Agentic-PRs.

TABLE II: Task- and Agent-Specific Quality Patterns

Dimension	Best Performer	Insight
Task Type	Docs (0.701, 84%)	Agents excel at structured, descriptive tasks.
CI/CD	0.675 (79%)	High automation success in build and test workflows.
Bug Fix	0.587 (65%)	Requires human contextual knowledge.
Agent Type	Codex (0.69, 82.6%)	Most reliable and consistent outputs.
Others	0.52–0.59	Lower consistency or domain specialization.

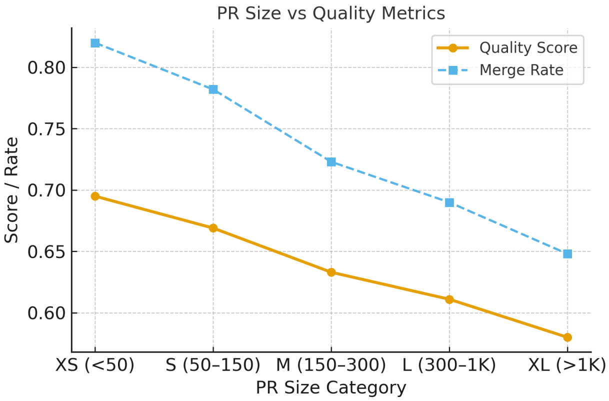

Fig. 2: PR size vs. quality score. Merge probability declines sharply beyond 300 lines.

best on documentation (84% merge rate), CI/CD (79%), and formatting tasks (77.7%), while debugging and algorithmic refactoring remain challenging (65% for bug fixes). Codex consistently outperforms other models in both merge probability and reviewer approval rate, producing smaller PRs (median 142 lines) compared to other agents (267–524 lines).

2) PR Size, Commit Granularity, and Quality: **Finding 3:** PR size shows strong negative correlation with quality ($r = -0.064$). Small PRs (<50 lines) achieve 82% merge rates versus 64.8% for large PRs (>1K lines), with medium effect size ($d = 0.36$).

TABLE III: Structural Correlates of Agentic-PR Quality (33,596 PRs)

Factor	Optimal Practice	Quantitative Effect
PR Size	<150 lines	Quality 0.67 vs 0.61; 82% merge ($p < .001$, $d = .36$).
Commits	Single commit	Quality 0.68 vs 0.54; $r = -0.18$.
Tests Included	Yes (44%)	+7.7 pp merge; -70% change requests.
Description Length	200–500 chars	Highest quality (0.69); $r = -0.21$.
Commit Message	>100 chars	Top predictor (48.9% importance).

Finding 4: Commit granularity exhibits unexpected patterns. Single-commit PRs achieve highest quality (0.677) and merge rates (79.5%), contradicting traditional atomic-commit practices. This suggests AI agents generate more complete, cohesive solutions, whereas multiple commits may indicate iterative debugging or uncertainty.

Finding 5: Test inclusion dramatically improves quality. PRs with tests show 76% merge rate versus 68.3% without tests ($p < .001$, $d = 0.159$). Test presence reduces change requests by 70% (from 0.067 to 0.020). However, test inclusion varies by task: 89.8% for test tasks, 50.1% for features, only 30.7% for bug fixes.

Finding 6: Communication quality outweighs technical complexity. Random Forest analysis reveals commit message length (48.9% importance) and PR description length (18.5%) as top predictors, exceeding technical factors like total changes (10.1%). Medium-length descriptions (200–500 chars) achieve optimal quality (0.687, 81.7% merge), while minimal or excessive descriptions correlate with lower acceptance.

V. Implications for Practice

PR quality in AI-assisted development depends primarily on disciplined structural practices rather than technical complexity. Analysis of 33,596 PRs demonstrates how empirical patterns can inform concrete guidelines:

- **PR size management:** Changes below 150 lines correspond to 17.2 pp higher merge rates (82% vs 64.8%).
- **Commit strategy:** Single-commit PRs achieve 23.3 pp higher merge rates, suggesting cohesive agent outputs outperform iterative ones.
- **Testing practices:** Test inclusion yields +7.7 pp merge and -70% change requests, highlighting value of automated test generation.
- **Documentation:** Commit message length and description quality are the strongest predictors of PR success.
- **Task selection:** Agents perform best on structured documentation and CI/CD tasks; human oversight remains vital for bug fixing.

VI. Conclusion

Across nearly one million PRs, AI coding agents are adopted broadly across experience levels and programming ecosystems, primarily by experienced practitioners (43.3%) using small repositories in Python-centric environments. PR quality depends less on technical scope and more on disciplined, communicative practices: maintaining small changes ($<$150 lines), single commits, test inclusion, and concise documentation (200–500 chars). Task type significantly affects success rates, with documentation and infrastructure tasks achieving highest quality (84%, 79%) while bug fixes face challenges (65%). Codex currently leads in both adoption (87.3%) and output quality (82.6% merge rate). These results show AI agents evolving from assistive tools to active collaborators in software engineering.

References

[1] P. Vaithilingam *et al.*, "Towards more effective ai-assisted programming," in *Proceedings of the 45th International Conference on Software Engineering (ICSE 2023)*, 2023, accessed: October 2025. [Online]. Available: https://glassmanlab.seas.harvard.edu/papers/Vaithilingam2023ICSE_IntelliCode.pdf

[2] J. T. Liang, C. Yang, and B. A. Myers, "A large-scale survey on the usability of ai programming assistants: Successes and challenges," *arXiv preprint arXiv:2303.17125*, 2023, accessed: October 2025. [Online]. Available: https://arxiv.org/abs/2303.17125